海量实时广告流平台架构设计与实践（一）

1. 平台背景概述
2. 平台流程详解
3. 平台痛点深度剖析
4. 平台架构设计与架构选型
5. 集群资源评估

海量实时广告流平台架构设计与实践（二）

1. 数据幂等性方案设计与实践
2. 冷热数据方案设计与实践
3. 实时数据修复方案设计与实践
4. 数据事件时间归档方案设计与实践

海量实时广告流平台架构设计与实践（三）

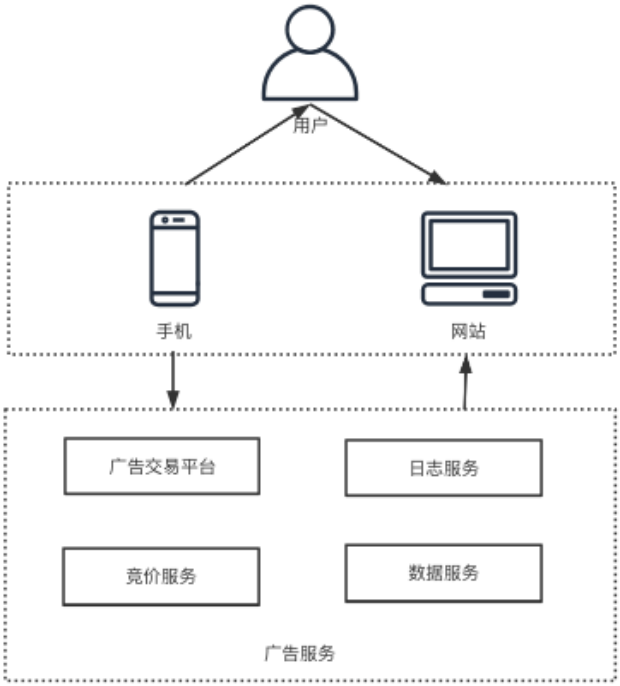
1. 数据重复方案设计与实践
2. 时间方案归档设计与实践
3. 离线数仓功能实现
4. Druid数据聚合方案实现
5. BI报表展示方案设计

课前准备

我们奈学提供的预操作/准备（集群运行环境）

- Flink
- Druid
- Hbase
- Hive
- Redis

项目背景（重要）



需求分析（重要）

1. 广告主要看实时统计广告效果数据，并且支持多维度分析
2. 把服务端广告竞价阶段的信息补充到客户端上报的日志里，需要实时日志拼接
3. 还有最基本的保证数据的准确性，需要有离线流程定时做数据修复

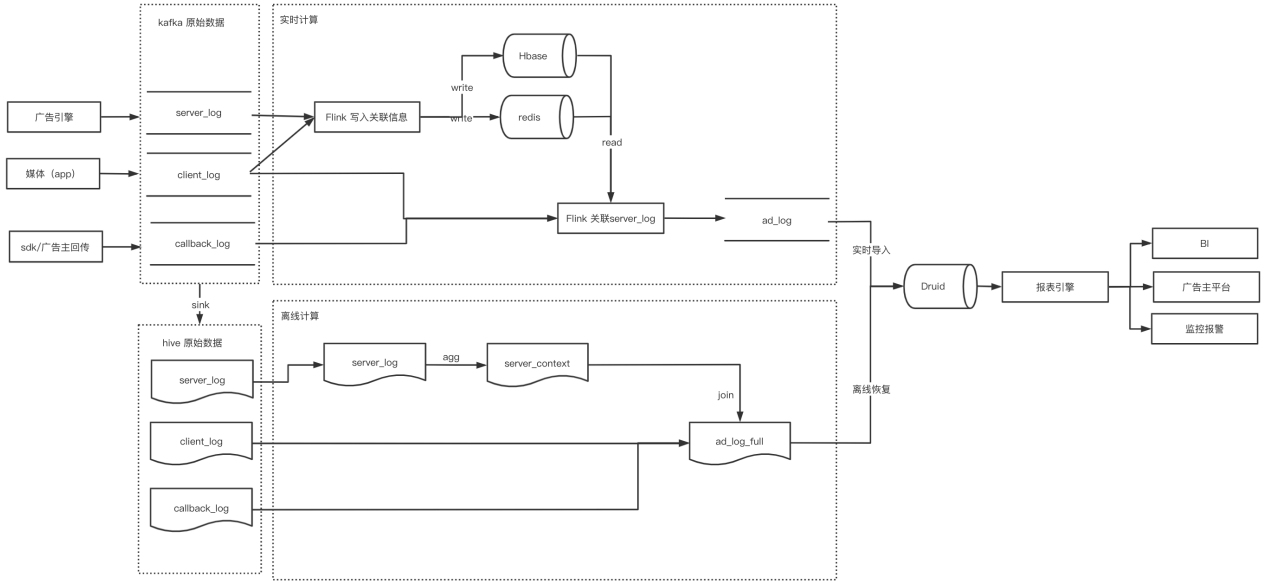
架构选型（重要）

- 技术选型(1) 方案一：
  - 广告请求时把信息编码到广告url里面， 曝光和点击的时候再上报。
  - 缺点：
    - 广告url过长，“偷”用户流量，移动端优质APP难以接受。
    - 不能放入过多的信息。
- 技术选型(2) 方案二：
  - 在Flink里面实现流式关联
  - 缺点：
    - join 窗口的双流数据都是被缓存在内存中的，也就是说如果某个key上的窗口数据太多就会导致 JVM OOM。
    - 广告下发后的曝光，转化等数据可能需要几天之后才能回传回来，内存无法缓存如此长时间的请求数据。
- 技术选型(3) 方案三：
  - 广告请求信息写入redis和hbase，广告曝光和点击时查询hbase，做日志关联
  - 优点：
    - redis里保存近两个小时的数据，大部分的查询能命中 cache
    - hbase里保存近1个月的数据，作为兜底逻辑

架构设计（重要）

数据收集 --> 数据预处理 -->数仓建模(分层设计 事实表 维度建模) --> 需求案例实现

LAMBDA



资源评估（重要）

高峰期数百万级别的QPS，数GB每秒的流量吞吐

Kafka：数百个partition

Flink：数千core

集群部署

提供部署方案. 奈学帮助实现!

代码实现（重要）

代码测试 + 项目运行

项目总结

本项目提供了一套完整的数据处理流程，通过lambda架构来保证数据的准确性

答疑

8:00 到 10:00 - 11:00